

# MAD2 PROJECT REPORT

## IESCP - V2

**Name:** Sarthak Singh Gaur

**Roll No:** 21f3001936

**Email:** 21f3001936@ds.study.iitm.ac.in

### About Me:

I am currently pursuing Bachelor of Science in Data Science and Applications, and I've reached the Diploma level of my studies. Throughout this journey, I've consistently honed my technical skills and expanded my knowledge base. A significant milestone in this process has been my work on an Application Development Project, which has been instrumental in sharpening my coding abilities and deepening my understanding of key technologies.

### Description:

This project is a web application built using the VueJS and Flask-Restful framework. It includes user registration and role-based redirection functionality with dashboard for each role. CRUD functionalities are implemented for Campaigns and Ad requests, with role based access to different entities and an option to update profile, campaigns and ad requests. The admin can flag all the entities and see the stats of the app. Daily and monthly reminders and report are also provided using Redis and Celery via mail and G-chat webhooks.

### Technologies Used:

VueJS (CLI): Frontend of the application.

Flask: Backend/API development.

Flask-Restful: To develop the Restful API for the app.

Flask-Security: For Token based authentication.

Sqlite3 Flask-SQLAlchemy: Simplified database interaction with ORM.

Redis and Celery are used for scheduled jobs/daily reminders via Google Chat and Mail

Flask-Mail: For sending monthly report emails.

Flask-CORS: To enable CORS for the app.

Flask-Caching: For caching and API performance optimisation

Jinja2: For generating monthly activity reports at backend.

SQLite: Easy-to-use SQL database engine for data storage.

### Database Schema Design:

#### 1. User Table:

**id** as primary key, a unique and required **username** and **email**, with a hashed **password**, **role** for their user rights and their activity **status**, using SQLAlchemy in python. The admin role is only added manually when the first request is sent and not through sign up page for security reasons.

#### 2. Influencer Table:

**id** as the primary key, **user\_id** as foreign key to the user table, with **name**, **category**, **niche**, **reach** and **platform** fields. Also a **flagged** field is there to identify suspicious influencers all using SQLAlchemy in python.

3. **Sponsor Table:** **id** as the primary key, **user\_id** as foreign key to the user table, with **company\_name**, **industry** and **budget** fields along with **flagged** field to identify suspicious sponsors using SQLAlchemy in python.
4. **Campaign Table:** **id** as primary key, **name**, **description**, **start** and **end** dates, **budget**, **visibility** and **goals** along with **sponsor\_id** as foreign key using SQLAlchemy in python.
5. **AdRequest Table:** **id** as primary key, **name**, **messages**, **requirements**, **pament** amount and **status** as fields for **adrequest**. It also has **sponsor\_id**, **campaign\_id** and **influencer\_id** as foreign key to sponsor table, campaign table and influencer table using SQLAlchemy in python.

## API Design

CRUD on Campaign/Ad Requests:

Description: Campaign/Ad-Requests API allows performing CRUD operations on Campaign/Ad Requests. It supports HTTP methods like GET, POST, PUT, and DELETE to manage each campaign and ad request created by sponsors in the database.

- API for User Login, Signup, Sponsor and Influencer Registration and Logout
- API for Sponsor and Influencers to search, manage and negotiate Ad Requests
- API for Admin to monitor, flag suspicious users, campaigns and approve Sponsors before allowing them on the platform

These API were implemented using Flask to define resources. SQLAlchemy was used to interact with the database and the data is returned as JSON responses by Flask's jsonify.

## Architecture and Features:

The application follows the MVC structure:

Model(M) is handled by flask - It interacts with databases and manages data. View(V) is handled by vue.js. Vue components are responsible for interactive user interface. Controller(C) is handled by flask. Flask Resources handle all the business logic at the backend.

## Project Features Implemented:

- Role based Login for Admin, Sponsors and Influencers
- Admin can see the app statistics, flag users, campaigns and ad requests
- Influencers can search public Campaign/Ad Requests and Negotiate the terms
- Sponsors can create Campaign/Ad Requests and search for Influencers
- Monthly Activity Report and Campaign report sent to Admin and Sponsor Email IDs
- Daily notifications on google chat to Influencers to visit the app if inactive for 24 hours.

**Video Link:** To view the Demo - [VIDEO DEMO](#)